Assessing the generational capabilities of Retrieval Augmented Systems using fine-tuned embedding model.

Dhiraj Pimparkar dpimpark@purdue.edu
Fahad Mehfooz fmehfooz@purdue.edu
Prantar Borah pborah@purdue.edu
Sarth Kulkarni kulka153@purdue.edu

Abstract

Retrieval Augmented Generative (RAG) systems have emerged as pivotal tools in natural language processing, facilitating enhanced information retrieval and generation tasks. However, their effectiveness hinges on the ability to accurately retrieve relevant information, particularly when confronted with domain-specific documents. This challenge is exacerbated by the limited pre-training of embedding models on such domains. Previous research[1] suggests that fine-tuning embedding models can ameliorate retrieval performance in RAG systems. Yet, scant attention has been paid to whether this enhancement extends to the generation phase. This study addresses this gap by leveraging fine-tuning techniques on embedding models and evaluating the resultant impact on the generative capabilities of RAG systems. Leveraging a diverse dataset, we conduct extensive experiments to assess the efficacy of the new RAG system in comparison to conventional RAG systems across various retrieval scenarios. Our findings shed light on the potential of fine-tuning embedding models to augment both retrieval and generation processes within RAG systems, thereby advancing their utility in real-world applications.

1 Introduction

In recent years, Retrieval Augmented Generative (RAG) systems have emerged as pivotal tools in the domain of natural language processing, offering a unique blend of information retrieval and generation capabilities. These systems hold immense promise in addressing a wide array of real-world challenges, from assisting with information synthesis to enhancing question answering tasks. However, their efficacy hinges on the ability to accurately retrieve relevant information, particularly in domain-specific contexts where conventional approaches may falter.

One of the primary challenges faced by RAG systems lies in their inability to retrieve accurate information when confronted with domain-specific documents. This limitation is often exacerbated by the embeddings generated by pre-trained models, which may not always align optimally with the specific objectives of retrieval and generation tasks. While these embeddings may exhibit proximity or disparity based on their pre-training objectives, they may not fully capture the nuances required for effective retrieval and generation within a given domain. Previous research[2] has indicated that fine-tuning embedding models could potentially address this issue and improve retrieval performance in RAG systems. However, the extent to which this improvement translates into enhanced generative capabilities remains largely unexplored.

A primary focus on retrieval, to the exclusion of generation, introduces certain limitations and assumptions. Firstly, retrieval-centric approaches assume that the relevant information is neatly contained within the retrieved "chunks" of text. However, real-world scenarios often involve information spanning across multiple chunks or even absence altogether, leading to incomplete or inaccurate answers. Additionally, in the absence of relevant information, generative LLMs [3] may attempt to fabricate responses based on context, potentially resulting in partially correct or wholly incorrect responses. Finally, the presence of irrelevant or distracting text within the retrieved chunks can further impede the generative LLM's ability to produce accurate outputs, a phenomenon highlighted[4] in previous research regarding LLMs' susceptibility to noise and bias from extraneous information.

In light of these considerations, this study aims to investigate the potential of fine-tuning embedding models to augment both retrieval and generation processes within RAG systems. Using a diverse data set[5] and conducting comprehensive experiments, we seek to evaluate the efficacy of a new RAG system compared to conventional approaches in various retrieval scenarios. Through this exploration, our aim is to advance our understanding of how fine-tuning embedding models can enhance the capabilities of RAG systems and pave the way for their broader adoption in practical applications.

2 Research Question

How does fine-tuned embedding models impact the retrieval and generation capabilities of Retrieval Augmented Generative (RAG) systems, particularly in domain-specific contexts?

3 Dataset

To facilitate the evaluation of retrieval capabilities, we conducted web scraping of Uber and Lyft's annual reports, utilizing one as training data and the other as validation data. This allowed us to assess whether fine-tuning the embedding model resulted in improved retrieval performance, particularly in domain-specific contexts.

For evaluating the generation capabilities of the new RAG system, we employed the PatronusAIFinance Benchmark dataset[6] obtained from Llamaindex[7]. It is derived from *FinanceBench* dataset comprising a diverse set of 10,000 question-answer pairs derived from publicly available financial documents such as SEC filings, earnings reports, and earnings call transcripts. We selected a subset, consisting of 150 annotated examples designed to evaluate and analyze various models' performance in financial question answering tasks. These questions cover a wide range of scenarios and are intended to serve as a minimum performance standard for assessing RAG systems' generative capabilities.

The FinanceBench dataset 1 spans several key areas of financial knowledge, including numerical reasoning, information retrieval, logical reasoning, and world knowledge. This comprehensive benchmark enables a thorough evaluation of the new RAG system's ability to handle diverse financial scenarios and provide accurate and insightful responses. Through the use of these data sets, we aim to rigorously assess the performance of the new RAG system and provide valuable insights into its potential applications in real-world settings.

4 Experimental Setup

4.1 Models Used

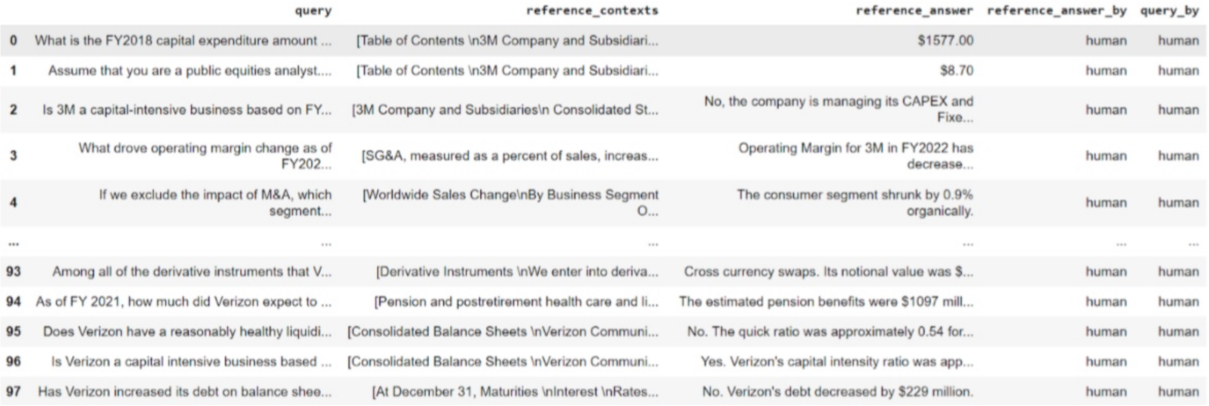
a. Embedding Model (BGE): The embedding model employed in this study is the BAAI General Embedding (BGE) model, accessible on HuggingFace. Developed by the Beijing Academy of Artificial Intelligence (BAAI), the BGE model stands out as a leading open-source embedding model. It is pre-trained using retromae[8] and fine-tuned on large-scale paired data using contrastive learning techniques. We selected this model primarily due to its open-source nature, as other embedding models, such as the OpenAI embedding model, are not available for fine-tuning.

b. Synthetic Dataset Simulation: A synthetic dataset was generated for fine-tuning purposes using the ChatGPT 3.5-turbo model API[9]. This dataset consists of question-answer pairs, where the answers correspond to segments of the dataset, and the questions are generated based on these segments using the ChatGPT 3.5-turbo model API. This approach facilitates the creation of synthetic positive pairs of (query, relevant documents) at scale, without the need for human annotators. Furthermore, within the RAG system, the ChatGPT 3.5-turbo model was employed to generate responses to queries based on retrieved context. Serving as the language model component of the RAG system, this model enables the generation of contextually relevant and coherent responses.

4.2 Fine-Tuning the Embedding Model:

Retrieval Augmented Generative (RAG) systems are commonly divided into two primary stages: indexing and query processing. In the indexing phase, raw textual data undergoes preparation, parsing, and segmentation before being embedded into a vectorized database. During query processing, relevant context is retrieved from this vector database using top-k embedding similarity search[9] and subsequently fed into the language model (LLM) for further processing.

When fine-tuning embeddings, it's essential to have training samples, typically comprising both "positive" and "negative" pairs of texts representing close and distant relationships, respectively. However, obtaining such examples beforehand poses a challenge. Thus, we use LlamaIndex modules to automatically generate a set of questions from unstructured text chunks. These (question, chunk) pairs are then used as positive examples as training signals for the model (negative examples are



	query	reference_contexts	reference_answer	reference_answer_by	query_by
0	What is the FY2018 capital expenditure amount ...	[Table of Contents \n3M Company and Subsidiari...	$1577.00	human	human
1	Assume that you are a public equities analyst....	[Table of Contents \n3M Company and Subsidiari...	$8.70	human	human
2	Is 3M a capital-intensive business based on FY...	[3M Company and Subsidiaries\n Consolidated St...	No, the company is managing its CAPEX and Fixe...	human	human
3	What drove operating margin change as of FY202...	[SG&A, measured as a percent of sales, increas...	Operating Margin for 3M in FY2022 has decrease...	human	human
4	If we exclude the impact of M&A, which segment...	[Worldwide Sales Change\nBy Business Segment O...	The consumer segment shrunk by 0.9% organically.	human	human
...	...	...	...	...	...
93	Among all of the derivative instruments that V...	[Derivative Instruments \nWe enter into deriva...	Cross currency swaps. Its notional value was $...	human	human
94	As of FY 2021, how much did Verizon expect to ...	[Pension and postretirement health care and li...	The estimated pension benefits were $1097 mill...	human	human
95	Does Verizon have a reasonably healthy liquidi...	[Consolidated Balance Sheets \nVerizon Communi...	No. The quick ratio was approximately 0.54 for...	human	human
96	Is Verizon a capital intensive business based ...	[Consolidated Balance Sheets \nVerizon Communi...	Yes. Verizon's capital intensity ratio was app...	human	human
97	Has Verizon increased its debt on balance shee...	[At December 31, Maturities \nInterest \nRates...	No. Verizon's debt decreased by $229 million.	human	human

Figure 1: FinanceBench Dataset (Human Annotated)

randomly sampled across other chunks). We followed the following steps to fine-tune the embedding model:

1. We first process the given documents into a corpus of text chunks. We do this with the SimpleNodeParser module in LlamaIndex.

2. Then for each text chunk, we use LLM to generate a few hypothetical questions that can be answered with information from that text chunk. The example prompt is shown below as well.

3. Finally, we collect all pairs of questions and text chunks as the dataset.

4. We leverage the high-level model fitting API from sentencetransformers to very easily setup a training process. We use MultipleNegativesRankingLoss as the training objective to fine-tune the embedding model.

5. We fine-tune for 3 epochs on the synthetic question-answer pair dataset.

4.3 Retrieval Evaluation:

To conduct retrieval evaluation, Lyft's annual reports for the year 2022 serve as the training data, while Uber's annual reports for the same period are utilized for validation purposes. This dataset choice facilitates evaluation in a domain-specific context, enhancing the relevance of the assessment to real-world scenarios.

Two main metrics are employed for evaluation:

1. **Hit Rate:** This metric determines the presence of relevant documents in the top-k retrieved documents for each (query, relevant_doc) pair. A hit is recorded if the results contain the relevant document, indicating successful retrieval.[10]

2. **InformationRetrievalEvaluator:**This comprehensive suite of metrics from sentence_transformers includes MAP (mean average precision), cosine similarity accuracy, precision, and recall at different top-k values[11]. By utilizing this evaluator, a detailed understanding of the retrieval performance can be gained across various retrieval scenarios and thresholds.

Through rigorous evaluation using these metrics, the effectiveness of the fine-tuned embedding model in improving retrieval capabilities within RAG systems can be quantified and compared against the base model.

4.4 Generation Evaluation:

In this study, generation evaluation focuses on comparing the performance of the new RAG system against the base RAG system, leveraging the PatronusAIFinance Benchmark dataset obtained from Llamaindex.

LabelledRagDataset[12] is primarily used for testing the performance of a RAG system by first generating a predicted (or generated) response to the given query using an LLM (ChatGPT 3.5 turbo model (using API)) and then comparing it to the reference answer. For convenience, we have a LlamaPack called the RagEvaluatorPack that streamlines this evaluation process!

RagEvaluatorPack[13] contains the following evaluation measures:

1. **Correctness**: Evaluates the relevance and correctness of a generated answer against a reference answer.

Model	Hit Rate	MAP
Naive BGE/BAAI	0.864	0.692
Fine-Tuned BGE/BAAI	0.890	0.741

Table 1: Retrieval Evaluation: Performance Metrics - Hit Rate and MAP

Model	Accuracy@1	Accuracy@3	Accuracy@5	Accuracy@10
Naive BGE/BAAI	0.578	0.767	0.833	0.888
Fine-Tuned BGE/BAAI	0.627	0.824	0.890	0.934

Table 2: Retrieval Evaluation: Accuracy Metrics

Model	Precision@1	Precision@3	Precision@5	Precision@10
Naive BGE/BAAI	0.578	0.256	0.167	0.089
Fine-Tuned BGE/BAAI	0.627	0.275	0.178	0.093

Table 3: Retrieval Evaluation: Precision Metrics

Model	Recall@1	Recall@3	Recall@5	Recall@10
Naive BGE/BAAI	0.578	0.767	0.833	0.888
Fine-Tuned BGE/BAAI	0.627	0.824	0.890	0.934

Table 4: Retrieval Evaluation: Recall Metrics

2. **Relevancy**: Measures if the query was actually answered by the response.

3. **Faithfulness**: Measures if the response was hallucinated.

4. **Context Similarity**: Measures the semantic similarity between the reference contexts and the contexts retrieved by the RAG system to generate the predicted response.

The file generated by the RagEvaluatorPack contains mean scores of all the above metrics and provides comprehensive details about the generation capabilities of RAG systems. We compare these metrics between the base RAG pipeline and our new pipeline. This evaluation serves to validate the effectiveness of the fine-tuned embedding model in enhancing the generative performance of RAG systems in domain-specific contexts.

5 Experimental Results

The experimental results are presented in two sections, evaluating the performance of the RAG system in terms of retrieval and generation capabilities.

The first set of results 1, 2, 3, 4 pertains to the retrieval performance of the RAG system. The tables display metrics such as hit rate, mean average precision (MAP), accuracy at various cutoffs, precision at different cutoffs, and recall at different cutoffs for both the naive BGE/BAAI model and the fine-tuned BGE/BAAI model.

The second set of results 5 focuses on the generation performance of the RAG system. The tables present mean scores for various evaluation metrics, including correctness, relevancy, faithfulness, and context similarity, for both the Naive BGE/BAAI model and the fine-tuned BGE/BAAI model.

6 Discussion

The experimental results shed light on the efficacy of fine-tuning the embedding model in improving the performance of the Retrieval Augmented Generative (RAG) system.

The retrieval evaluation metrics demonstrate notable enhancements in the performance of the Fine-Tuned BGE/BAAI model compared to the Naive BGE/BAAI model. The hit rate and mean average precision (MAP) show an improvement from 0.864 and 0.692, respectively, for the Naive model to 0.890 and 0.741 for the Fine-Tuned model. This indicates that fine-tuning the embedding model has resulted in better retrieval of relevant context, leading to improved precision in answering queries.

Further analysis of accuracy, precision, and recall at various cutoff points reaffirms the superior

Model	Mean Correctness Score	Mean Relevancy Score	Mean Faithfulness Score	Mean Context Similarity Score
Naive BGE/BAAI	3.533	0.306	0.602	0.849
Fine-Tuned BGE/BAAI	3.592	0.347	0.622	0.849

Table 5: Generation Evaluation: Mean Scores

performance of the Fine-Tuned BGE/BAAI model. Across all metrics, including accuracy@1, precision@1, and recall@1, the Fine-Tuned model consistently outperforms its Naive counterpart. This suggests that the fine-tuning process has positively impacted the RAG system's ability to retrieve relevant information and provide accurate responses, particularly at higher cutoff values.

In terms of generation performance, the mean scores for correctness, relevancy, faithfulness, and context similarity exhibit marginal improvements in the Fine-Tuned BGE/BAAI model compared to the Naive model. While the differences in mean scores are relatively small, they indicate a slight enhancement in the quality and coherence of the generated responses. Notably, both models demonstrate relatively high scores for correctness and context similarity, indicating the overall effectiveness of the RAG system in generating relevant and contextually appropriate responses.

7 Conclusion

In conclusion, this study investigated the impact of fine-tuning the embedding model on the performance of Retrieval Augmented Generative (RAG) systems. Through comprehensive evaluation of retrieval and generation capabilities, we have demonstrated the effectiveness of fine-tuning in enhancing the overall functionality of RAG systems.

The experimental results revealed significant improvements in retrieval performance, as evidenced by higher hit rates, mean average precision (MAP), accuracy, precision, and recall metrics for the Fine-Tuned BGE/BAAI model compared to the Naive BGE/BAAI model. This enhancement in retrieval capabilities indicates the importance of fine-tuning in facilitating more accurate and relevant information retrieval, leading to improved precision in generating responses to user queries.

Furthermore, the generation evaluation results showed marginal but discernible improvements in the quality and coherence of generated responses for the Fine-Tuned BGE/BAAI model. While the differences in mean scores were relatively small, they underscore the positive impact of fine-tuning on the generation performance of RAG systems, particularly in producing contextually relevant and accurate responses.

Overall, the findings of this study highlight the potential of fine-tuning embedding models to enhance the effectiveness and practical applicability of RAG systems in real-world natural language processing tasks. By leveraging fine-tuning techniques, researchers and practitioners can optimize RAG systems for domain-specific contexts, improving their ability to retrieve and generate relevant information effectively.

Looking ahead, future research endeavors could explore additional fine-tuning strategies, alternative embedding models, and ensemble techniques to further augment the performance of RAG systems. Additionally, ongoing efforts to integrate user feedback and refine RAG architectures could contribute to the continual advancement and refinement of these systems, ultimately empowering them to address complex natural language processing challenges with greater accuracy and efficacy.

8 Reference

https://arxiv.org/html/2404.07221v1

[1] LLAMA Index Blog. Fine-tuning embeddings for rag with synthetic data, 2023. https://medium.com/llamaindex-blog/fine-tuning-embeddings-for-rag-with-synthetic-data-e5

[2] Rangan, K. (2024). A Fine-tuning Enhanced RAG System with Quantized Influence Measure as AI Judge. https://arxiv.org/html/2402.17081v1

[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language

Models are Unsupervised Multitask Learners. https://arxiv.org/abs/1901.02860

[4] Havrilla, A., & Iyer, M. (2024). Understanding the Effect of Noise in LLM Training Data with Algorithmic Chains of Thought. https://arxiv.org/abs/2402.04004

[5] Lyft Annual Report https://s27.q4cdn.com/263799617/files/doc_financials/2023/ar/LYFT-ANNUAL-REPORT-2023.pdf

[6] PatronusAIFinance Benchmark dataset https://huggingface.co/datasets/PatronusAI/financebench

[7] Llama Index Website https://www.llamaindex.ai/

[8] Xiao, S., Liu, Z., Shao, Y., & Cao, Z. (2022). RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder https://arxiv.org/abs/2205.12035

[9] ChatGPT Turbo 3.5 https://platform.openai.com/docs/models/gpt-3-5-turbo

[10] Medium Article- Embedding similarity search https://medium.com/@kvrware/embedding-similarity-search-25c6911240af

[11] Information retrieval and top-K Github repo https://github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers/evaluation/InformationRetrievalEvaluator.py

[12] Benchmarking RAG Pipelines With A LabelledRagDatatset https://docs.llamaindex.ai/en/stable/examples/llama_dataset/labelled-rag-datasets/

[13] RAG Evaluator Pack- Llama Index https://docs.llamaindex.ai/en/stable/api_reference/packs/rag_evaluator/

9 Team Members and Contributions

- Dhiraj Pimparkar: Developed scripts for fine-tuning and conducted evaluations for both retrieval and generation. Presented the project in class and authored the report.

- Sarth Kulkarni: Assisted with creating the PowerPoint presentation and creating report. Helped develop scripts for fine-tuning in generation and gave mid-week presentation

- Fahad Mehfooz:

- Prantar Borah:

10 GitHub Repository

The code for this project can be found on GitHub at the following link:

https://github.com/d29parkar/Generational-Assessment-RAGs